REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

Université Larbi Ben M'hidi Oum el Bouaghi
ISTA Ain M'lila
Département Réseaux et Télécommunications

1ère Année Licence
Module : Informatique 1

Numération & codage

© Dr BENACER Imad Jan 2021

Année Universitaire : 2020 / 2021

PLAN D'EXPOSE

- Introduction
- Système décimal
- Système binaire , octal et hexadécimal
- Conversion d'un système de numération vers un autre système .
- Opérations arithmétiques en binaire, octal et hexadécimal.

Objectifs

- Comprendre c’est quoi un système de numération .
- Apprendre la méthode de conversion d’un système à un autre .
- Apprendre à faire des opérations arithmétiques en binaire.

Introduction

Les informations traitées par l'ordinateur sont de différents types, quelle que soit la nature de l'information traitée par un ordinateur (image, son, texte, vidéo), ces informations sont toujours représentées sous forme d'un ensemble de nombres écrits en base 2 ou binaire, par exemple 01001011.

Un système de numération est un ensemble de règles permettant de représenter les nombres. Les systèmes de numération font correspondre à un nombre N un certain symbolisme dit forme polynomiale.

Introduction

- Nous avons pris l'habitude de représenter les nombres en utilisant dix symboles différents: 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9
- Ce système est appelé le système décimal (déci signifie dix).
- Il existe cependant d'autres formes de numération qui fonctionnent en utilisant un nombre de symboles distincts.
  - Exemple :
    - système binaire (bi: deux),
    - le système octal (oct: huit),
    - le système hexadécimal (hexa: seize).
- En fait, on peut utiliser n'importe quel nombre de symboles différents (pas nécessairement des chiffres).
- Dans un système de numération : le nombre de symboles distincts est appelé la base du système de numération.

Introduction

Dans un système de numérotation en base B, un nombre noté $N_{(B)}$ égal à (forme polynomiale):

$$\boldsymbol{N_{(B)}} = \sum_{k=0}^{n-1} \mathrm{a}_k B^k = \mathbf{a_k\, B^k + a_{k\text{-}1}\, B^{k\text{-}1} + \ldots + a_1\, B^1 + a_0\, B^0}$$

Avec :

B : base ou nombre de chiffres différents qu'utilise le système de numérotation.

a_k : Chiffre de rang k

1 . Le système décimal

- On utilise dix symboles différents:

  { 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 }

- N'importe quelle combinaison des symboles **{ 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 }** nous donne un nombre.

$\boxed{2}\ 3\ 3\ 4\ 5\ 6\ \boxed{7}$

Poids fort

Poids faible

$\boxed{\mathbf{345}}\,,\,\boxed{\mathbf{567}}$

Partie entière

Partie fractionnelle

Développement en polynôme d'un nombre dans le système décimal

- Soit le nombre 1978, ce nombre peut être écrit sous la forme suivante :

$$1978 = 1000 + 900 + 70 + 8$$

$$1978 = 1*1000 + 9*100 + 7*10 + 8*1$$

$$1978 = 1*10^3 + 9*10^2 + 7*10^1 + 8*10^0$$

Cette forma s'appelle la forme polynomiale

Un nombre réel peut être écrit aussi sous la forme polynomiale

$$1978{,}265 = 1*10^3 + 9*10^2 + 7*10^1 + 8*10^0 + 2*10^{-1} + 6*10^{-2} + 5*10^{-3}$$

2 . Système binaire (système à base 2)

- Dans le système binaire, pour exprimer n'importe quelle valeur on utilise uniquement 2 symboles : { 0 , 1 }

La base

Un bit $(\boxed{1}101)_2$

$(\boxed{1}\ 1\ 0\ \boxed{1})_2$

Le bits du poids forts

Le bits du poids faible

. Un nombre dans la base 2 peut être écrit aussi sous la forme polynomial

$$(1110)_2 = 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 = (14)_{10}$$

$$(1110{,}101)_2 = 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} = (14{,}625)_{10}$$

Comptage en binaire

- Sur un seul bit : 0 , 1

.Sur 2 bits :

4 combinaisons= 2^2

Binaire	Décimal
00	0
01	1
10	2
11	3

Sur 3 Bits   8 combinaisons= 2^3

Binaire	Décimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

- en général le nombre des chiffres qu'en peut les coder est 2^n

L'intervalle de codage est de 0 à 2^n-1

Le système octal (base 8)

- 8 symboles sont utilisés dans ce système:

$$\{ 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 \}$$

- **<u>Exemple 1 :</u>**

$$(127)_8 = 1*8^2 + 2*8^1 + 7*8^0$$

$$(127{,}65)_8 = 1*8^2 + 2*8^1 + 7*8^0 + 6*8^{-1} + 5*8^{-2}$$

<u>Exemple 2 :</u>
Le nombre (1289) n'existe pas dans la base 8 puisque les symboles 8 et 9 n'appartiennent pas à la base .

Le système hexadécimal (base 16)

- On utilise seize (16) symboles différents:

$$(17)_{16} = 1*16^1 + 7*16^0$$

$$(AB)_{16} = A*16^1 + B*16^0 = 10*16^1 + 11*1$$

Décimal	Hexadécimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Résumé

- Dans une base X , on utilise X symboles distincts pour représenter les nombres.
- La valeur de chaque symbole doit être strictement inférieur à la base X.
- Chaque nombre dans une base X peut être écrit sous sa forme polynomiale .

3. Conversion d'une base X à la base 10

- Cette conversion est assez simple puisque il suffit de faire le développement en polynôme de ce nombre dans la base X , et de faire la somme par la suite.

Exemple :

$$(1101)_2 = 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = (13)_{10}$$

$$(1A7)_{16} = 1*16^2 + A*16^1 + 7*16^0 = 1*16^2 + 10*16^1 + 7*16^0 = 256+160+7 = (423)_{10}$$

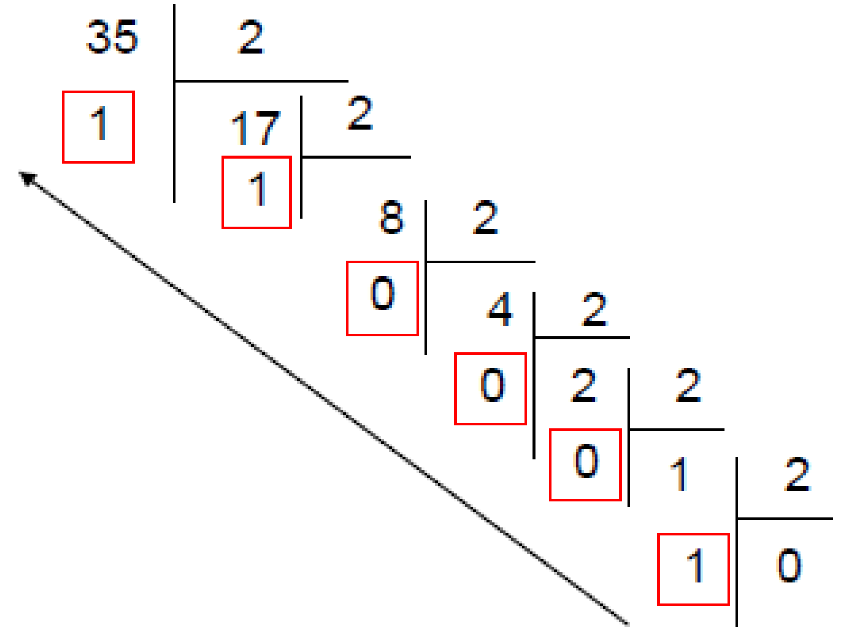
$$(1101,101)_2 = 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} = (13,625)_{10}$$

$$(43,2)_5 = 4*5^1 + 3*5^0 + 2*5^{-1} = 20+3+0,4 = (23,4)_{10}$$

Conversion de la base 10 à la base 2

Le principe consiste à faire des divisions successives du nombre sur 2 , et prendre le reste des divisions dans l'ordre inverse.

Exemple 1 : $(35)_{10}=(?)_2$

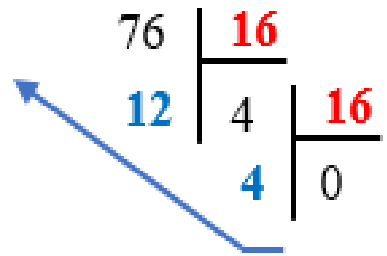


Après division :
on obtient : $(35)_{10}=(100011)_2$

➤ Conversion décimal - hexadécimal

Pour transférer de la base décimale vers une base 16, on applique la méthode de division successive décrites déjà auparavant.

Exemple :

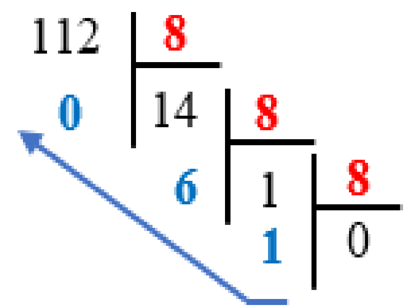


Convertir $(76)_{10}$ en hexadécimal (base B=16).

76 s'écrit donc en base 16 : **4C**.

➤ Conversion décimal - octal

Le schéma ci-contre explique la méthode mieux qu'un long discours :



Exemple :

Convertir $(112)_{10}$ en octal (base B=8).

112 s'écrit donc en base 8 : **160**.

Conversion de la base 10 à la base 2 : cas d'un nombre réel

- Un nombre réel est constitué de deux parties : la partie entière et la partie fractionnelle.
- La partie entière est transformée en effectuant des divisions successives.
- La partie fractionnelle est transformée en effectuant des multiplications successives par 2 .

Exemple : $35{,}625=(?)_2$

P.E= $35 = (100011)_2$

PF= $0{,}625 = (?)_2$

$0{,}625 * 2 = \boxed{1},25$

$0{,}25 * 2 = \boxed{0},5$

$0{,}5 * 2 = \boxed{1},0$

$(0{,}625)=(0{,}101)_2$

Donc $35{,}625=(100011{,}101)_2$

- **Exemple 2:** $(0,6)_{10}=(?)_2$

$0,6 * 2 = 1,2$

$0,2 * 2 = 0,4$

$0,4 * 2 = 0,8$

$0,8 * 2 = 1,6$

$\longrightarrow$ $(0,6) = (0,1001)_2$

Remarque :

Le nombre de bits après la virgule va déterminer la précision .

Exemple 3

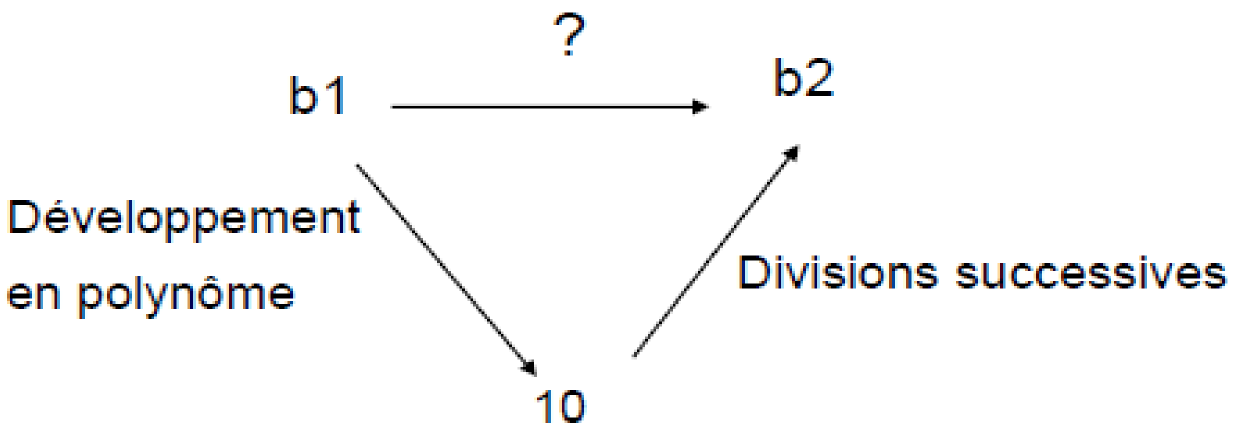
Convertir 0,732 en octal (base B=8).

$0,732 * 8 = 5,856$
$0,856 * 8 = 6,848$
$0,848 * 8 = 6,784$
$0,784 * 8 = 6,272$
$0,272 * 8 = 2,176$

On remarque ici qu'on ne peut pas avoir une partie fractionnaire nulle qu'après un grand nombre de multiplication, c'est pour ça qu'on arrête au cinquième chiffre, on obtient alors : $(0,732)_{10} = (0,56662)_8$.

Conversion d'une base b1 à une base b2

- Il n'existe pas de méthode pour passer d'une base b1 à une autre base b2 directement.
- L'idée est de convertir le nombre de la base b1 à la base 10 , en suit convertir le résultat de la base 10 à la base b2 .



Conversion : Octal → binaire

. En octal chaque, symbole de la base s'écrit sur 3 bits en binaire.
. L'idée de base est de replacer chaque symbole dans la base octal par sa valeur en binaire sur 3 bits (faire des éclatement sur 3 bits).

Octal	Binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Exemples :

$(345)_8=(\underline{011}\ \underline{100}\ \underline{101})_2$

$(65,76)_8=(\underline{110}\ \underline{101},\ \underline{111}\ \underline{110})_2$

$(35,34)_8=(\underline{011}\ \underline{101}\ ,\ \underline{011}\ \underline{100})_2$

Conversion : binaire → octal

- L'idée de base est de faire des regroupements de 3 bits à partir du poids faible.
- Par la suite remplacer chaque regroupement par la valeur octal correspondante .

Exemple :

$(11001010010110)_2=(\underline{011}\ \underline{001}\ \underline{010}\ \underline{010}\ \underline{110})_2=(31226)_8$

$(110010100,10101)_2= (\underline{110}\ \underline{010}\ \underline{100}\ ,\ \underline{101}\ \underline{010})_2=(624,51)_8$

Remarque :
le regroupement se fait de droit à gauche pour la partie entière et de gauche à droite pour la partie fractionnelle .

Conversion : hexadécimal → binaire

. En Hexa chaque symbole de la base s'écrit sur 4 bits.

. L'idée de base est de replacer chaque symbole par sa valeur en binaire sur 4 bits (faire des éclatement sur 4 bits).

Décimal	Hexadécimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Exemple :

$(345B)_{16}=(\underline{0011}\ \underline{0100}\ \underline{0101}\ \underline{1011})_2$

$(AB3,4F6)_{16} = (\ \underline{1010}\ \underline{1011}\ \underline{0011}\ ,\ \underline{0100}\ \underline{1111}\ \underline{0110}\ )_2$

Conversion : binaire →hexadécimal

. L'idée de base est de faire des regroupements de 4 bits à partir du poids faible.

Par la suite remplacer chaque regroupement par la valeur Héxa correspondante .

Exemple :

$(11001010100110)_2$=$(\underline{0011}\ \underline{0010}\ \underline{1010}\ \underline{0110})_2$=$(32A6)_{16}$

$(110010100{,}10101)_2$= $(\underline{0001}\ \underline{1001}\ \underline{0100}{,}\underline{1010}\ \underline{1000})_2$=$(194{,}A8)_{16}$

4. Opérations arithmétiques en binaire

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 1\ 0 \end{array}$$

Exemple

$(15)_{10} = (1111)_2$

$(9)_{10} = (1001)_2$

$$\begin{array}{r} 15 \\ +\ 09 \\ \hline 24 \end{array} \qquad \begin{array}{r} \color{red}{1\ 1\ 1\ 1} \\ 1\ 1\ 1\ 1 \\ +\ 1\ 0\ 0\ 1 \\ \hline 1\ 1\ 0\ 0\ 0 \end{array}$$

$$\begin{array}{r} \color{red}{1}\ \ \color{red}{1}\ \ \\ 1\ 1\ 0\ 0\ 0\ 1\ 1 \\ +\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0 \end{array}$$

Multiplication binaire

se base sur les quatre opérations suivantes

1x 1 = 1 0 x 1 = 0
1 x 0 = 0 0 x 0 = 0

$$
\begin{array}{lr}
 & 1001 \\
 & \cdot\ \ 1011 \\
\hline
= & 1001 \\
 & 1001 \\
 & 0000 \\
 & 1001 \\
\hline
= & 1100011
\end{array}
$$

multiplicande = $(9)_{10}$

multiplicateur = $(11)_{10}$

produits partiels

produit final = $(99)_{10}$

Soustraction binaire

Elle repose sur les quatre opérations suivantes:

$$0 - 0 = 0$$
$$1 - 0 = 1$$
$$0 - 1 = 1 \quad \text{on emprunt 1}$$
$$1 - 1 = 0$$

Exemple

$(19)_{10} = (10011)_2$

$(14)_{10} = (01110)_2$

$$\begin{array}{r} 19 \\ - \quad 14 \\ \hline 05 \end{array}$$

$$\begin{array}{r} 1\ 0\ 0\ 1\ 1 \\ 1\ 1 \\ -\quad 0\ 1\ 1\ 1\ 0 \\ \hline 0\ 0\ 1\ 0\ 1 \end{array}$$

NB : la soustraction s'effectue de la même manière dans les autres bases.

Division binaire:

La division d'un nombre binaire (le dividende) par un autre (le diviseur) est identique à la division de deux nombres décimaux.

Exemple

$(29)_{10} = (11101)_2$

$(5)_{10} = (101)_2$

$$
\begin{array}{r}
29 \\
\div \quad 5 \\
\hline
5 \text{ et le reste } 4
\end{array}
$$

$$
\begin{array}{rr|l}
 & 11101 & 101 \\
\cline{3-3}
- & 101 & 1\ 0\ 1 \\
\cline{2-2}
 & 0100 & \\
- & 000 & \\
\cline{2-2}
 & 1001 & \\
- & 101 & \\
\cline{2-2}
 & 0100 &
\end{array}
$$

Opérations arithmétiques en octal

	$\boxed{1}$	$\boxed{1}$		
	4	3	6	5
+		4	5	1
	$\boxed{5}$	$\boxed{8}$	$\boxed{11}$	$\boxed{6}$

En octal 8 s'écrit 10

En octal 11 s'écrit 13

$\boxed{0}$ $\boxed{3}$

Le résultat final : $(\mathbf{5036})_8$

Opérations arithmétiques en hexadécimal

	1			
	4	8	6	5
+	7	A	5	1
	12	18	11	6

C

En hexa 18 s'écrit 12

2

En hexa 11 s'écrit B

B

Le résultat final : $(\mathbf{C2B6})_{16}$

5. Quel est le système utilisé dans les dispositifs numériques ?

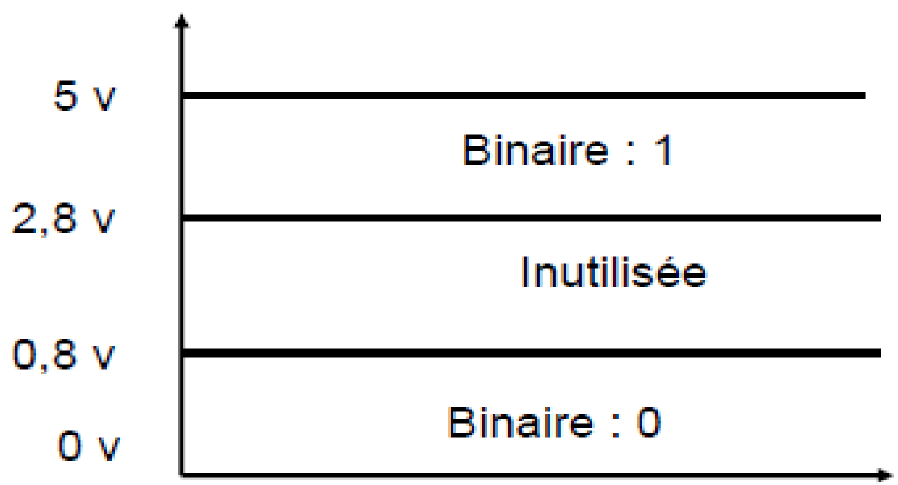
. Les machines numériques utilisent le système binaire.

. Dans le système binaire : uniquement 2 symboles sont utilisés : 0 et 1.

. C'est facile de représenter ces deux symboles dans les machines numériques.

. Le 0 et le 1 sont représentés par deux tensions .

Binaire (logique)	Tension
0	0 V
1	5 V



6. Représentation des nombres signés

Les nombres sont soit positifs, soit négatifs, le signe (+) ou (-) est identifié par un bit, dit le **bit de signe** et le nombre ainsi formé est dit signé. Lorsque le bit de signe est à 0 il désigne un nombre positif et lorsqu'il est à 1 il désigne un nombre négatif (0 : positif, 1 : négatif).

Généralement, il est mis à gauche (position de poids fort), mais il n'entre pas dans l'évaluation du nombre.

- $(+19)_{10}$ = **0** 0010011 : le bit de poids fort à gauche (**0**) est le bit de signe (+)
- $(-19)_{10}$ = **1** 1101101 : le bit de poids fort à gauche (**1**) est le bit de signe (-)

On peut identifier trois principales façons de représenter les nombres négatifs (***complément à deux***) :

1. Écrire la valeur absolue du nombre en base 2. Le bit de poids fort doit être égal à 0.

2. Inverser les bits : les 0 deviennent des 1 et vice versa. On fait ce qu'on appelle le complément à un.

3. On ajoute 1 au résultat.

<u>Complément à un & Complément à deux</u>

Valeur	*Complément à 1*
000	111
111	000
101	010

Valeur	*Complément à 2*
001	111
011	110
010	010

Exemple:

On désire coder la valeur -19 sur 8 bits. Il suffit :

1. d'écrire 19 en binaire : 00010011

2. d'écrire son complément à 1 : 11101100

3. et d'ajouter 1 : 11101101

La représentation binaire de -19 sur 8 bits est donc 11101101.

***NB* : Lorsqu'en additionnant un nombre et son complément à deux on obtient 0.**

Dans l'exemple précédent : 00010011 + 11101101 = 0000000 (avec une retenue de 1 qui est éliminée).

- Intervalle de valeurs représentées pour les entiers relatifs : $[-2^{n-1}, 2^{n-1} - 1]$

- 4 bits : on peut représenter les entiers de [-8 à 7]
- 8 bits : on peut représenter les entiers de [-128 à 127]
- 16 bits : on peut représenter les entiers de [-32768 à 32767]

- Intervalle de valeurs représentées pour les entiers positifs : $[0, 2^{n} - 1]$

- 4 bits : on peut représenter les entiers de [0 à 15]
- 8 bits : on peut représenter les entiers de [0 à 255]
- 16 bits : on peut représenter les entiers de [0 à 65535]

Notion de codage

Le processus de codage permet de transformer une matière brute (données collectées) en un ensemble d'éléments pertinents interprétables.

On appelle codage l'opération qui consiste à établir une loi de correspondance des informations à représenter (lettre, chiffre, signe,...) appelée code par un symbole où un ensemble de symboles particuliers (configurations binaires) dans le but de produire une information plus explicite et/ou plus aisément exploitable. De plus, chaque information correspond une et une seule configuration binaire, et une configuration binaire est attribuée à une seule information.

L'opération de codage est de présenter chaque information par un code. L'opération de décodage c'est l'inverse, à partir d'un code on trouve l'information représentée par ce code.

Exemples de codes

- Le code des mois : janvier = 1, février = 2,… décembre = 12.

Pourquoi commencer en janvier ?

Si tout le monde utilise le même code, tout le monde comprend que "9" veut dire "septembre"

- Les codes postaux, les numéros de sécurité sociale,..

Tout le peuple algérien comprend que "05000" veut dire "Batna", "04000" veut dire "Oum El Bouaghi"

Lorsque l'on parle des informations codées, deux types de catégories sont abordés, les premiers sont les données numériques qui pouvant être l'objet d'une opération numérique et les seconds les données non numériques ou alphanumériques (symboles constituant un texte).

➢ Données numériques

Les données numériques sont de différents types :

- Entiers naturels : 0, 1, 2, …
- Entiers relatifs (positifs et négatifs) : -2, -1, 0, 1, 2, …
- Nombres fractionnaires : 10,5.
- Nombres en notation scientifique : 4,5x10e3.

➢ Données non numériques

Les données non numériques correspondent aux caractères alphabétiques A… Z et a… z, aux dix chiffres 0… 9 et aux caractères spéciaux ! + - * / ; ? : …

Données numériques

a- Code binaire naturel

Le code binaire naturel (BN) est la succession naturelle des nombres binaires. Or cette succession est illimitée il faudra donc préciser la taille des nombres afin d'être précis.

Le code binaire naturel et ses dérivés (octal et hexadécimal) répondent aux règles classiques de l'arithmétique des nombres positifs.

Exemple :

$(7)_{10}$ sur 8 bits = $(0000\ 0111)_2$

$(10)_{10}$ sur 8 bits = $(0000\ 1010)_2$

Données numériques

b- Code BCD (Binary Coded Decimal)

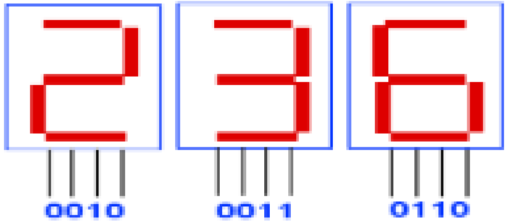
Ce code **BCD**, en Français **DCB** (Décimal Codé Binaire) est un système de numération utilisé en électronique et en informatique pour coder des nombres en se rapprochant de la représentation humaine usuelle, en base 10.

BCD consiste à représenter chaque chiffre d'un nombre décimal par son équivalent binaire sur 4 bits.

<u>Exemples</u>

Pour coder un nombre tel que 127, il suffit de coder chacun des chiffres 1, 2 et 7 séparément :

1	**2**	**7**	**Code décimale**
↓	↓	↓	
0001	0010	0111	**Code BCD**



Ce codage ne permet aucun calcul, il est uniquement destiné à la saisie et à l'affichage de données

c- Code Gray (code binaire réfléchi)

Le code binaire réfléchi (BR) ou code de Gray est un code utilisé pour ses propriétés de symétrie dans le codage de position ou pour la simplification des équations logiques (tableaux de Karnaugh). Le principe consiste à changer l'état d'un seul bit entre deux nombres consécutifs comme le montre le tableau ci-dessous.

Le terme réfléchi est dû à la symétrie qui apparaît dans le code.

Binaire pur

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Binaire réfléchi

0	0	0	0
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	1	1
0	1	0	1
0	1	0	0
1	1	0	0
1	1	0	1
1	1	1	1
1	1	1	0
1	0	1	0
1	0	1	1
1	0	0	1
1	0	0	0

Symétrie

Données non numériques (Codes alphanumériques ou codage des caractères)

a- Code ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Le code ASCII (ASCII = American Standard Code for Information Interchange) permet à toutes sortes de machines de stocker, analyser et communiquer de l'information textuelle. La norme ASCII (on prononce généralement « aski ») établit une correspondance entre une représentation binaire des caractères de l'alphabet latin et les symboles, les signes, qui constituent cet alphabet. Par exemple, le caractère a est associé à 1100001 (97) et A à 1000001 (65).

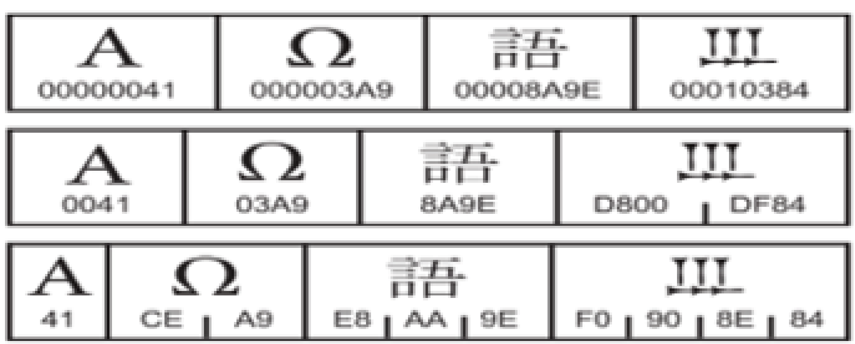
- 48 à 57 : chiffres dans l'ordre (0,1,…,9)
- 65 à 90 : les alphabets majuscules (A,…,Z)
- 97 à 122 : les alphabets minuscule (a,…z)

Le code ASCII de base représentait les caractères sur 7 bits (c'est à- dire 128 caractères possibles, de 0 à 127). Il englobe des lettres, des chiffes, des signes de ponctuations et un certain nombre de signaux de commande. Toutes ces correspondances sont fixées par l'American National Standards Institutes.

c- UNICODE

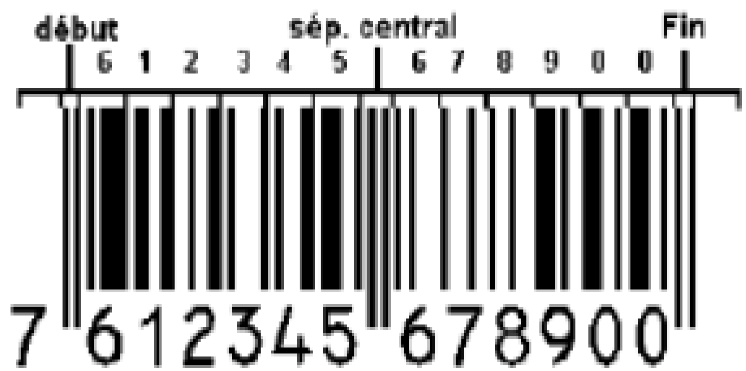
Il existe d'autres normes que l'ASCII, comme l'Unicode par exemple, qui présentent l'avantage de proposer une version unifiée des différents encodages de caractères complétant l'ASCII. Il comprend les signes de presque toutes les langues écrites du monde. Y compris ceux qui ne s'appliquent plus. Arabes, japonais, coréen, écriture maya, et alphabets d'anciens états. La notation des mesures et des poids, la notation musicale, les concepts mathématiques sont présentés.

Unicode est une norme de codage de caractères. Autrement dit, il s'agit d'une table de correspondance de caractères textuels (chiffres, lettres, éléments de ponctuation) avec des codes binaires. Unicode définit des dizaines de milliers de codes, mais les 128 premiers restent compatibles avec ASCII.



e- Codes-barres EAN

Le numéro EAN (European Article Numbering) identifie des articles ou des unités logistiques de façon unique, codé sous forme de codes-barres. L'EAN est composé de 8 ou 13 chiffres représentés sous forme de séquences de barres noires et blanches formant un code-barres. Ce type de code-barres se trouve sur la presque totalité de produits courants (alimentation, vêtements, droguerie, papeterie, électroménager, etc.). Il peut indiquer le pays d'origine, le nom du fabricant, celui du produit, sa référence.



f- Codes QR

Le code QR (ou QR code en anglais) est un code-barres en deux dimensions (ou code à matrice) constitué de modules noirs disposés dans un carré à fond blanc. Le nom QR est l'acronyme de l'anglais Quick Response, car son contenu de données peut être décodé rapidement.

Le code QR a été créé par l'entreprise japonaise Denso-Wave en 1994 pour le suivi des pièces de voiture dans les usines de Toyota.

Les codes QR peuvent mémoriser des adresses web, du texte, des numéros de téléphone, des SMS ou autres types de données lisibles par les smartphones et les téléphones mobiles équipés d'une application de lecture (lecteur de code QR ou QR reader en anglais).

Merci pour votre aimable attention